Introduction to security and cryptography
Lecture 4: Asymmetric cryptography

Marius Lombard-Platet

ENS, Almerys

Today

Asymmetric encryption

- Mathematical hard problems
- RSA-OAEP, ElGamal, Diffie-Hellman
- Attack resistance
- Signatures

Hard problems for asymmetric crypto

Behind asymmetric crypto lies the concept of one-way functions.

One-way functions are easy to compute, hard to invert.

Hard problems for asymmetric crypto

Behind asymmetric crypto lies the concept of one-way functions.

One-way functions are easy to compute, hard to invert.

Fun fact: we do not know whether one-way functions exist (as this would imply $P \neq NP$). But we have good candidates.

Example: factorization

Given two prime numbers, it is easy to compute their product

But given a semi-prime number, it is hard to compute its two prime factors.

Exercise: factor

- 15

Example: factorization

Given two prime numbers, it is easy to compute their product

But given a semi-prime number, it is hard to compute its two prime factors.

Exercise: factor

- 15
- 143

Example: factorization

Given two prime numbers, it is easy to compute their product

But given a semi-prime number, it is hard to compute its two prime factors.

Exercise: factor

- 15
- 143
- a real RSA 2048 key: 25 195 908 475 657 893 494 027 183 240 048 398 571 429 282 126 204 032 027 777 137 836 043 662 020 707 595 556 264 018 525 880 784 406 918 290 641 249 515 082 189 298 559 149 176 184 502 808 489 120 072 844 992 687 392 807 287 776 735 971 418 347 270 261 896 375 014 971 824 691 165 077 613 379 859 095 700 097 330 459 748 808 428 401 797 429 100 642 458 691 817 195 118 746 121 515 172 654 632 282 216 869 987 549 182 422 433 637 259 085 141 865 462 043 576 798 423 387 184 774 447 920 739 934 236 584 823 824 281 198 163 815 010 674 810 451 660 377 306 056 201 619 676 256 133 844 143 603 833 904 414 952 634 432 190 114 657 544 454 178 424 020 924 616 515 723 350 778 707 749 817 125 772 467 962 926 386 356 373 289 912 154 831 438 167 899 885 040 445 364 023 527 381 951 378 636 564 391 212 010 397 122 822 120 720 357 (maybe don't)

RSA (Rivest-Shamir-Adleman)

Let p and q be two *big* distinct prime numbers, of similar size.

Public key $pk = (n, e)$

- $n = p \cdot q$
- $\gcd(e, \phi(n)) = 1$ (with $\phi(n) = (p-1)(q-1)$)

Secret key $sk = d$

- $e \cdot d \equiv 1 \mod \phi(n)$

RSA / Encryption and Decryption

Encryption
Let M be the message to encrypt, then

$$C \equiv M^e \mod n$$

RSA / Encryption and Decryption

Encryption
Let M be the message to encrypt, then

$$C \equiv M^e \mod n$$

Decryption
Let C be the message to decrypt, then

$$M \equiv C^d \mod n$$

Complexity Estimates

Estimates for integer factoring [Lenstra-Verheul 2000]

Modulus n (bits)	Operations ($\log_2$)	
512	58	
1024	80	$\approx 2^{60}$ years
2048	111	
4096	149	
8192	156	

Notions of security

How do we know it is secure?

Let's focus on confidentiality. How do we define secure?

How do we know it is secure?

Let's focus on confidentiality. How do we define secure?

It must be hard for an attacker (what kind of attacker?) to get information (what kind of information?) from the ciphertext.

Attack model

- Black box: the attacker is not the one doing the decryptions
- White box: under research, no satisfying results for the moment (for now it's too easy to pwn whitebox crypto)

The attacker will play an **adversarial game**. During this game, they will have to solve a **challenge**.

Defense model: the challenge

Attack reach:

- **Key Recovery** (KR): the attacker must retrieve the secret key.

Defense model: the challenge

Attack reach:

- **Key Recovery** (KR): the attacker must retrieve the secret key.
- **One-Way** (OW): the attacker must retrieve the cleartext from a ciphertext.

Defense model: the challenge

Attack reach:

- **Key Recovery** (KR): the attacker must retrieve the secret key.
- **One-Way** (OW): the attacker must retrieve the cleartext from a ciphertext.
- **Indistinguishability** (IND): attacker must identify from which cleartext of their choice belongs a ciphertext.

Defense model: the challenge

Attack reach:

- **Key Recovery** (KR): the attacker must retrieve the secret key.
- **One-Way** (OW): the attacker must retrieve the cleartext from a ciphertext.
- **Indistinguishability** (IND): attacker must identify from which cleartext of their choice belongs a ciphertext.
- **Non-malleability** (NM): the attacker modifies a ciphertext in a way that results in a still valid corresponding cleartext

Defense model: the challenge

Attack reach:

- **Key Recovery** (KR): the attacker must retrieve the secret key.
- **One-Way** (OW): the attacker must retrieve the cleartext from a ciphertext.
- **Indistinguishability** (IND): attacker must identify from which cleartext of their choice belongs a ciphertext.
- **Non-malleability** (NM): the attacker modifies a ciphertext in a way that results in a still valid corresponding cleartext

NM-secure $\Rightarrow$ IND-secure $\Rightarrow$ OW-secure $\Rightarrow$ KR-secure.

Exercise (tricky): exhibit a KR-secure cryptosystem which is not OW-secure.

Why isn't OW enough?

One-wayness means that the adversary cannot retrieve the whole message. But maybe they can retrieve half of the message!

Why isn't OW enough?

One-wayness means that the adversary cannot retrieve the whole message. But maybe they can retrieve half of the message!

Let's take a physical example. Consider a shredder: the paper is completely torn, we cannot read the message anymore (one-wayness).

However, we can determine whether the paper was red or white.

We gained one bit of information about the paper. Maybe this bit of information is critical!

IND challenge

Alice is the attacker, and she sends chooses two messages of her choice, m_0 and m_1. She sends them to the challenger.

The challenger selects a random bit b, and returns to Alice $Enc(m_b)$

Alice has to guess b with probability higher than $1/2$.

IND consequence

An IND secure scheme cannot be deterministic.

Hence, IND-secure schemes add randomness in the ciphertext: $Enc(m)$ will never give twice the same output. But $Dec(Enc(m))$ will always give m.

Attack model

Attacks in the black-box security model, from weaker to stronger:

- **Known-plaintext attack** (KPA): attacker knows a few pairs of cleartext-ciphertext. Not used in practice

Attack model

Attacks in the black-box security model, from weaker to stronger:

- **Known-plaintext attack** (KPA): attacker knows a few pairs of cleartext-ciphertext. Not used in practice
- **Chosen-plaintext attack** (CPA): attacker can ask the ciphertext of messages of their choice. This is always the case in a public key scheme.

Attack model

Attacks in the black-box security model, from weaker to stronger:

- **Known-plaintext attack** (KPA): attacker knows a few pairs of cleartext-ciphertext. Not used in practice
- **Chosen-plaintext attack** (CPA): attacker can ask the ciphertext of messages of their choice. This is always the case in a public key scheme.
- **Chosen-ciphertext attack** (CCA): attacker can ask for the ciphertexts of the message of their choice, and, before receiving the challenge, can ask for the cleartexts of the ciphertexts of her choice.

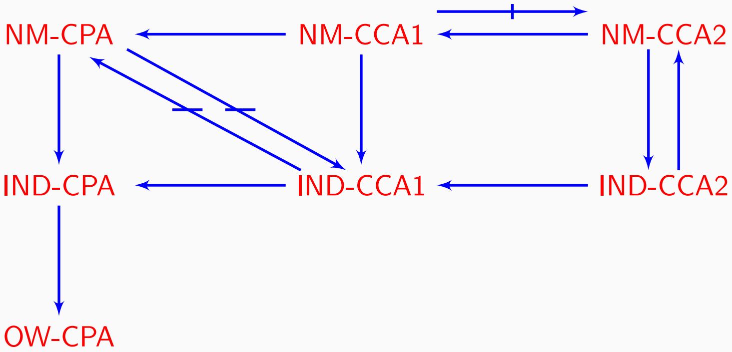
Attack model

Attacks in the black-box security model, from weaker to stronger:

- **Known-plaintext attack** (KPA): attacker knows a few pairs of cleartext-ciphertext. Not used in practice
- **Chosen-plaintext attack** (CPA): attacker can ask the ciphertext of messages of their choice. This is always the case in a public key scheme.
- **Chosen-ciphertext attack** (CCA): attacker can ask for the ciphertexts of the message of their choice, and, before receiving the challenge, can ask for the cleartexts of the ciphertexts of her choice.
- **Chosen-ciphertext attack 2** (CCA2): Same, but can do the same after having received the challenge (except asking to decrypt the challenge).

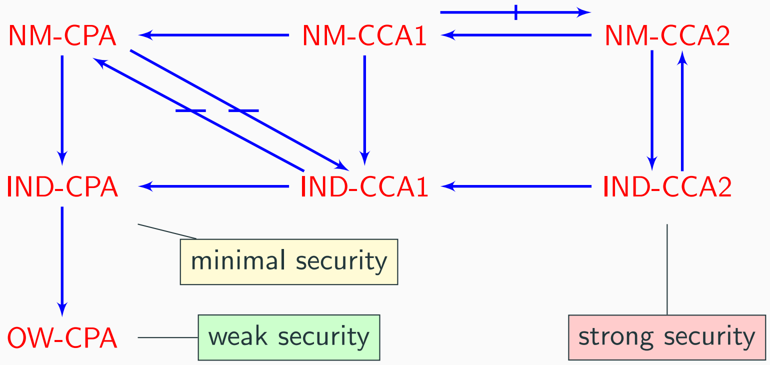
Note that CCA2 $\Rightarrow$ CCA resistance $\Rightarrow$ CPA resistance $\Rightarrow$ KPA.

Relations



"Relations Among Notions of Security for Public-Key Encryption Schemes", **Crypto'98**, by Mihir Bellare, Anand Desai, David Pointcheval and Phillip Rogaway [BDPR'98]

Relations



"Relations Among Notions of Security for Public-Key Encryption Schemes", **Crypto'98**, by Mihir Bellare, Anand Desai, David Pointcheval and Phillip Rogaway [BDPR'98]

Another Example of Algorithmically Hard Problem

Discrete Logarithm Problem
Let , g be a generator of the multiplicative group $\mathbb{Z}_p^*$ and $x \in \{1, \ldots, p-1\}$.

- Given p, g and x, it is **easy** to compute $y \equiv g^x \mod p$
- Given p, g and $y = g^x$, it is **difficult** to find x

This is not true in every group. For instance, p must be a safe prime number[1].

[1] A prime number is safe if $\frac{p-1}{2}$ is also prime.

ElGamal Encryption Scheme

Let p be a *big* safe prime number and g a generator of the multiplicative group $\mathbb{Z}_p^*$.

Private key $sk = x$

- $x \in \{1, \ldots, p-1\}$

Public key $pk = (g, p, h)$

- $h \equiv g^x \mod p$

ElGamal / Encryption and Decryption

Encryption
Let M be the message to encrypt and r a random element of $\{1, \ldots, p-1\}$, then

$$C = (C_1, C_2) = (g^r \mod p, M \cdot h^r \mod p)$$

ElGamal / Encryption and Decryption

Encryption
Let M be the message to encrypt and r a random element of $\{1, \ldots, p-1\}$, then

$$C = (C_1, C_2) = (g^r \mod p, M \cdot h^r \mod p)$$

Decryption
Let C be the cipher to decrypt, then

$$M \equiv C_2 \cdot C_1^{-x} \mod p$$

RSA vs ElGamal?

Exercise
Which is the difference between RSA and ElGamal if we encrypt the same message M twice?

OAEP (Optimal Asymmetric Encryption Padding)

Used with RSA, OAEP give the probabilistic property.

OAEP (Optimal Asymmetric Encryption Padding)

Used with RSA, OAEP give the probabilistic property.

Composition

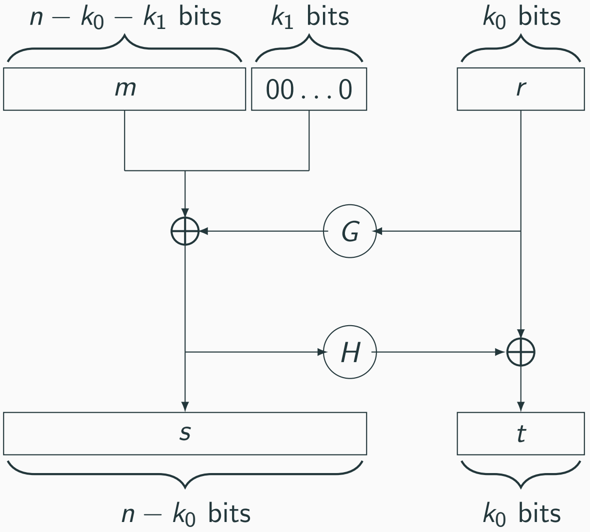
- A hash function G
- A hash function H
- Two XOR

OAEP (Optimal Asymmetric Encryption Padding)

Algorithm

1. m is padded with k_1 zeros to be $n - k_0$ bits in length
2. r is a randomly generated k_0-bit string
3. G expands the k_0 bits of r to $n - k_0$ bits
4. $s = m00\ldots0 \oplus G(r)$
5. H reduces the $n - k_0$ bits of s to k_0 bits
6. $t = r \oplus H(s)$

OAEP (Optimal Asymmetric Encryption Padding)



OAEP (Optimal Asymmetric Encryption Padding)

Exercise
What is the decryption algorithm of OAEP?

OAEP (Optimal Asymmetric Encryption Padding)

Decryption Algortithm

$$r = t \oplus H(s)$$

$$m = s \oplus G(r)$$

If $[m]_{k_1} = 0^{k_1}$, the algorithm returns $[m]^n$, otherwise it returns "Reject"

- $[m]_{k_1}$ denotes the k_1 least significant bits of m
- $[m]^n$ denotes the n most significant bits of m

Others Cryptosystems

- Bellare & Rogaway (1993)

$$f(r)||x \oplus G(r)||H(x||r)$$

- Zheng & Seberry (1993)

$$f(r)||G(r) \oplus (x||H(x))$$

Diffie-Hellman Key Exchange

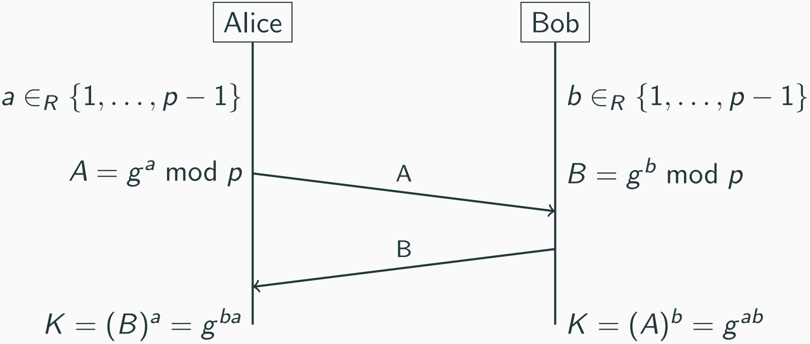
Idea
Use properties of asymmetric encryption to exchange secret key between Alice and Bob.

Diffie-Hellman's method is based on *Discret Logarithm Problem*.

Diffie-Hellman Key Exchange

Public parameters:

g, p



Digital Signature

Digital Signature

Definition
A *digital signature* is a mathematical scheme for demonstrating the authenticity of digital messages or documents.

A signature scheme depends on a asymmetric cryptosystem.

Digital Signature

Alice
(pk_A, sk_A)

Bob

$\text{sgn}_{sk_A}(m) = s \longrightarrow s$

Bob checks the signature with the Alice's public key.

Signature with RSA

Let p and q be two *big* prime numbers.

Public key $pk = (n, e)$

- $n = p \cdot q$
- $\gcd(e, \varphi(n)) = 1$

Secret key $sk = d$

- $e \cdot d \equiv 1 \mod \varphi(n)$

Signature
Let m be the message to sign, then

$$s \equiv m^d \mod n$$

RSA / Signature and Verification

Signature
Let m be the message to sign, then

$$s \equiv m^d \mod n$$

Verification
Let s be a signature, we verify the signature computing

$$m \stackrel{?}{\equiv} s^e \mod n$$

DSA: Digital Signature Algorithm

DSA is another signature scheme. For secure use, keys must be at least 2048 bits.

The 2048 bits requirement is because DSA relies on discrete logarithm problem in the group $(\mathbb{Z}/p\mathbb{Z})^*$.

However, in other groups, such as elliptic curves, the discrete logarithm is much harder.

Hence, ECDSA (Elliptic Curve DSA) only requires 256 bits keys.

Signatures in real life

The messages we want to sign can be huge.

Yet asymmetric crypto is slow.

Hence, a better idea: we sign $H(m)$, where H is a cryptographic function.

Security is the same, and because the hash is small (usually less than 512 bits), signatures are faster.

Asymmetric vs Symmetric

Comparison

- Size of the key
- Complexity of computation
- Key distribution
- Signature only possible with asymmetric scheme

Computational Cost of Encryption

2 hours of video (3Ghz CPU)

	DVD 4,7 G.B		Blu-Ray 25 GB	
Schemes	encrypt	decrypt	encrypt	decrypt
RSA 2048	22min	24h	115min	130h
RSA 1024	21min	10h	111min	53h
AES	20sec	20sec	105sec	105sec

Thank you for your attention

Any question?

Things to remember

- Difference between symmetric and asymmetric crypto
- IND-CPA model
- Factorisation problem, discrete log
- General description of Diffie-Hellman, RSA, OAEP
- Elliptic curves
- Signatures

Exercises

1. Sort from minimal security to maximal security: OW-CPA; IND-CPA; IND-CCA2
2. What are the differences between signatures and MAC?
3. In asymmetric crypto, can I give my secret key to anyone? Can I give my public key to anyone?
4. Is DSA-512 a secure algorithm? Is ECDSA-512 a secure algorithm?